7/PRTS

09/403560

1   420 Rec'd PCT/PTO 2 5 OCT 1999

DATA TRANSFER METHOD

FIELD OF THE INVENTION

The present invention relates to the field of digital data transfer methods,

5   more particularly to the transfer of data in which normal digital data and encrypted digital

data co-exist in the same data.

BACKGROUND OF THE INVENTION

One conventional data transfer method adopts the IEEE1394 standard

10   (IEEE: The Institute of Electrical and Electronics Engineers, Inc.). (Reference: IEEE Std

1394: 1995, High Performance Serial Bus.) In data transfer specified by the IEEE 1394

standard, there are two methods of communication. One is isochronous communication,

which is suitable for transferring synchronous data such as digital video signals and

digital audio signals. The other is asynchronous communication, which is suitable for

15   transferring asynchronous data such as control signals. Both methods of communication

are applicable on the IEEE 1394 bus network. Isochronous communication is what is

called "Broadcast communication, and an isochronous packet output from one device

coupled to the IEEE 1394 bus is receivable by all the other devices coupled to the same

bus. On the other hand, asynchronous communication is applicable to both one-to-one

20   communication and one-to-N broadcast communication. Each asynchronous packet output

from one device coupled to the bus contains an identifier specifying the device(s) to

which that packet is addressed. If this identifier specifies a particular device, only the

device specified by the identifier receives the asynchronous packet. If the identifier

specifies broadcast, all the devices coupled to the same bus receive the asynchronous

25   packet.

At present, the IEC (International Electrotechnical Commission) is

preparing to stipulate the IEC1883 standard (hereafter referred to as AV protocol) for

transferring digital audio signals and digital video signals or transmitting data between devices coupled to an IEEE 1394 bus, employing the data transfer method conforming to the IEEE 1394 standard. In the AV protocol, video and audio data is located in the isochronous packet as shown in Fig. 5 and transferred. The isochronous packet includes a

5   CIP (Common Isochronous Packet) header. The CIP header carries information that includes the type of AV data, the identification number of the device which is sending the isochronous packet, and the like.

Fig. 5 shows the format of the isochronous packet used in the AV protocol. The isochronous packet comprises an isochronous packet header 900, header CRC 901,

10   isochronous payload 902, and data CRC 903. The isochronous packet header 900 contains a tag 907. The tag 907 shows that the isochronous packet conforms to the AV protocol when its value is 1. When the value of the tag 907 is 1, which means that the isochronous packet conforms to the AV protocol, the isochronous payload 902 has a CIP header 904 at its beginning. The CIP header 904 comprises a source ID 906 which identifies the device

15   transmitting the isochronous packet. The CIP header 904 also comprises FMT 908 and FDF 909 which specify the type of actual data 905 in the isochronous payload 902. Digital AV data is contained in the actual data 905, but the actual data 905 is not always contained in the isochronous payload 902. Some packets may have an isochronous payload 902 which contains only the CIP header 904 without the actual data 905.

20   There is a group of commands called the AV/C Command Set for controlling devices in accordance with the AV protocol (Reference: 1394 TRADE ASSOCIATION Specification for AV/C Digital Interface Command Set Version 1.0, September 13, 1996). These commands and their responses are transferred by means of asynchronous communication.

25   In the conventional data transfer method as described above, compatibility with conventional devices which are not designed for transferring an encrypted

3

isochronous payload 902 cannot be secured when an encrypted isochronous packet, which contains the isochronous payload 902 which has been encrypted for copy protection, is sent. More specifically, conventional devices are designed with the precondition that the CIP header 904 is normally positioned at the beginning of the isochronous payload 902.

5   Accordingly, if the isochronous payload 902 is encrypted, conventional devices cannot correctly read out the encrypted CIP header 904, and decide that the isochronous packet does not conform to the AV protocol. A device receiving encrypted isochronous packets thus may not operate properly. In other words, such receiving devices cannot determine the type of data contained in the actual data 905, resulting in an inability to identify the

10   device transmitting the isochronous packet. In addition, asynchronous communication such as queries to the sending device are disabled. Accordingly, normal receiving operations cannot be carried out.

Furthermore, if the isochronous packet output from the sending device is encrypted while the receiving device is receiving the data, some conventional devices

15   may not be able to correctly read out the CIP header 904 as soon as encryption starts, resulting in inability to receive data properly.

In order to send AV information encrypted for copy protection from the sending device and decrypt the encrypted AV data by the authorized receiving device, the sending device needs to give decrypting information to the authorized receiving device. In

20   the conventional data transfer method, however, the sending device may be required to execute extremely complicated procedures in order to specify the receiving device. More specifically, each isochronous packet contains the source ID 906 which is the identifier of the sending device, but these packets do not contain information that identifies which device is authorized to receive these packets. The sending device thus cannot check which

25   device is receiving the isochronous packets during transmission of the isochronous packets. In order to find which of the devices coupled to the IEEE 1394 bus is receiving

the data, the sending device may require to query the data receiving status of every device coupled to the same bus. This makes the procedures for giving key information for decryption extremely complicated.

5 **SUMMARY OF THE INVENTION**

A data transfer method of the present invention satisfies the conventional communication standard even in the case of sending encrypted video and audio information via isochronous communication. In addition, the present invention offers a data transfer method which prevents erroneous operation even if conventional receiving

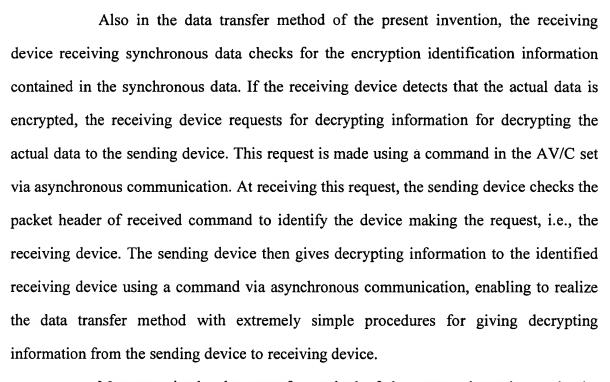10 devices receive isochronous packets containing encrypted video and audio data.

The present invention still further offers a data transfer method which significantly simplifies procedures for giving key information for decryption from a sending device to an authorized receiving device.

In a data transfer method of the present invention, synchronous data

15 transferred via isochronous communication contains i) encryption identification information which indicates encryption status of actual data and ii) actual data, and only the actual data is encrypted.

To solve another problem in the conventional data transfer method, the encryption identification information which indicates encryption status of the actual data

20 in the synchronous data is sent together with the actual data from the sending device so that receiving device can detect that the actual data is encrypted based on this encryption identification information and request decrypting information to the sending device in the data transfer method of the present invention. Then, the receiving device receiving the decrypting information sent from the sending device upon request decrypts the actual data

25 using this decrypting information to complete data transfer.

Also in the data transfer method of the present invention, the receiving device receiving synchronous data checks for the encryption identification information contained in the synchronous data. If the receiving device detects that the actual data is encrypted, the receiving device requests for decrypting information for decrypting the

5   actual data to the sending device. This request is made using a command in the AV/C set via asynchronous communication. At receiving this request, the sending device checks the packet header of received command to identify the device making the request, i.e., the receiving device. The sending device then gives decrypting information to the identified receiving device using a command via asynchronous communication, enabling to realize

10   the data transfer method with extremely simple procedures for giving decrypting information from the sending device to receiving device.

Moreover, in the data transfer method of the present invention, only the actual data in the synchronous data is encrypted, and the encryption identification information indicating the encryption status of the actual data is included in the

15   synchronous data. This enables to transfer the CIP header without being encrypted, preventing erroneous operation when the conventional device receives such encrypted synchronous data. In other words, the present invention realizes a data transfer method which assures compatibility with the conventional data transfer method and eliminates the possibility of erroneous operation when the conventional receiving device receives

20   encrypted synchronous data.

Furthermore, the data transfer method of the present invention eliminates the possibility of erroneous operation of the receiving device receiving data when encryption of synchronous data starts while continuously receiving synchronous data from the sending device because the CIP header is not encrypted and transferred as it is.

25
BRIEF DESCRIPTION OF THE DRAWING

Fig. 1 is a schematic view of a format of a CIP header in accordance with a preferred embodiment of the present invention.

Fig. 2 is a block diagram illustrating functions of sending and receiving devices in accordance with the preferred embodiment of the present invention.

5   Fig. 3A is a format of AKE status command in accordance with the preferred embodiment of the present invention.

Fig. 3B is a format of AKE response to the AKE status command in accordance with the preferred embodiment of the present invention.

Fig. 3C is a format of AKE control command in accordance with the

10 preferred embodiment of the present invention.

Fig. 4 is a schematic view illustrating procedures for transmitting an asynchronous packet between sending and receiving devices in accordance with the preferred embodiment of the present invention.

Fig. 5 is a format of isochronous packet in a data transfer method of the

15 prior art.

DESCRIPTION OF THE PREFERRED EMBODIMENT

A preferred embodiment of the present invention is described next with reference to drawings.

20 Fig. 1 shows a format of the payload of an isochronous packet to be transferred in the preferred embodiment of the present invention. The preferred embodiment is one example of the transfer of a TSP (Transport Packet) in accordance with MPEG (the Moving Picture Expert Group) specifications. The ENC (hereafter referred to as encryption status) 910 indicates whether the actual data 905 is encrypted or

25 not.

Fig. 2 shows the relation between sending and receiving devices in the preferred embodiment of the present invention. A sending device 110 and receiving device 128 are coupled via an IEEE 1394 bus (hereafter referred to as a 1394 bus) 111.

First, the functions of each block in the sending device 110 are described.

5   A signal source 100 outputs an MPEG transport packet TSP (not illustrated) in an 188 byte unit, which will be sent via the 1394 bus 111, to an encrypter 101. In other words, in the preferred embodiment, the signal source 100 outputs data with a fixed length of 188 bytes. The encrypter 101 encrypts and outputs the TSP received from the signal source 100 using an encryption key 109 provided by a key generator 106. In the

10   preferred embodiment, the encryption key is equivalent to the decrypting information. An output command 105 is a command from the key generator 106 to the encrypter 101. There are three types of commands: normal output, encrypted output, and empty output. If the encrypter 101 receives the output command 105 for normal output, the TSP received from the signal source 100 is output without modification, and registers the value 0 as the

15   encrypting information 910. If the output command 105 is for encrypted output, the encrypter 101 encrypts the TSP with the encryption key 109 received from the key generator 106, and registers the value 1 as the encrypting information 910. If the output command 105 is for empty output, the encrypter 101 outputs an empty signal (not illustrated) every time it receives a TSP from the signal source 100, and registers the

20   value 1 as the encrypting information 910. A source packet generator 102 adds a 4-byte source packet header to the 188-byte TSP received from the encrypter 101, and outputs a 192-byte source packet (actual data 905). A CIP block generator 103 adds a CIP header 954 to the source packet received from the source packet generator 102, and outputs an isochronous payload 952. Here, the CIP block generator 103 places the encrypting

25   information 910 received from the encrypter 101 in the CIP header 954. An isochronous packet generator 107 adds an isochronous packet header 900, header CRC 901, and data

CRC 903 to the isochronous payload 952 received from the CIP block generator 103, and outputs an isochronous packet. Since the content of the isochronous payload 952 is data that conforms to the AV protocol, the value of the tag 907 is set to 1. The key generator 106 sends the encryption key 109 to the receiving device 128 by communicating the

5   asynchronous packet with the receiving device 128, as shown in Fig. 3, which is described later. The key generator 106 also outputs the encryption key 109 to the encrypter 101 as described above.

A 1394 packet I/O means 108 inputs and outputs isochronous and asynchronous packets between the 1394 bus 111 and sending device 110. More

10   specifically, the 1394 packet I/O means 108 outputs the isochronous packet received from the isochronous packet generator 107 and asynchronous packet received from the key generator 106 to the 1394 bus 111, and also outputs asynchronous packet received from the 1394 bus 111 to the key generator 106.

Next, functions of each block of the receiving device 128 are described.

15   A 1394 packet I/O means 127 inputs and outputs isochronous and asynchronous packets between the 1394 bus 111 and receiving device 128. More specifically, the 1394 packet I/O means 127 outputs the isochronous packet received from the 1394 bus 111 to a payload extractor 123, and outputs asynchronous packet received from the 1394 bus 111 to a key generator 125. The 1394 packet I/O means 127 also

20   outputs asynchronous packet received from the key generator 125 to the 1394 bus 111.

The payload extractor 123 receives the isochronous packet, transmitted from the 1394 bus 111, from the 1394 packet I/O means 127. When the value of the isochronous packet tag 907 is 1, the payload extractor 123 determines that an isochronous payload 952 contains data conforming to the AV protocol, and outputs the isochronous

25   payload 952 to an actual data extractor 122. When received isochronous payload 952 contains the actual data 905, the actual data extractor 122 outputs the actual data 905 to a

decrypter 121, after removing the CIP header 954 placed at the beginning of the isochronous payload 952. The actual data extractor 122 also outputs the source ID 906 and encrypting information 910 extracted from the CIP header 954 to the key generator 125. The encrypting information 910 is also output to the decrypter 121. The key generator 125 receives an encryption key 126 as a result of exchanging asynchronous packet with the sending device 110 via asynchronous communication, which is described later, and outputs the encryption key 126 to the decrypter 121. When the value of the encrypting information 910 received from the actual data extractor 122 is 0, the decrypter 121 outputs the actual data 905 received from the actual data extractor 122 to an AV generator 120 as it is. When the value of the encrypting information 910 is 1, the decrypter 121 decrypts the actual data 905 using the encryption key 126 received from the key generator 125, and outputs decrypted actual data 905 to the AV generator 120.

Next, the transmission of an asynchronous packet via the aforementioned asynchronous communication setup is described.

Figs. 3A to 3C illustrate how the format of the asynchronous packet is transmitted by asynchronous communication. More specifically, Figs. 3A and 3C show the command formats of the AKE commands (AKE: Authentication and Key Exchange) communicated between the key generators 106 and 125. Fig. 3B shows the response format. These commands and responses belong to the AV/C Command Set, and are communicated between the sending device 110 and receiving device 128 using the asynchronous communication. By communicating these commands and responses, the sending device 110 and receiving device 128 exchange information required for the authentication of each other and encryption keys 109 and 126. The AKE commands comprise AKE control commands for requesting a target device to carry out a specific operation, and an AKE status commands for querying the status and capabilities of the target device.

Fig. 3A shows the format of the AKE status command. In the AKE status command, an operation code 208 indicates that this command is an AKE command. The value of the algorithm ID 200 is set at 0, with other values reserved for future extension.

Fig. 3B shows the format of responses to the AKE status commands. This is a response sent back from the device receiving the AKE status command to the device issuing the AKE status command. There are multiple procedures for exchanging information for mutual authentication and transmission of encryption keys 109 and 126 between the sending device 110 and receiving device 128. In an algorithm field 201, the identifier for an information exchange procedure which the device returning an applicable response can execute is assigned in bits. In other words, the receiving device 128 exchanges several commands and responses with the sending device 110 after an encrypted TSP is detected in line with the aforementioned procedures and before receiving the encryption keys 109 and 126. There is more than one procedure for communicating these commands and responses. The device sending back the response designates the executable information exchange procedure by setting 1 to an applicable bit in the algorithm field 201. Since the size of the algorithm field 201 is 16 bits, a maximum of 16 types of information exchange procedures can be indicated. The maximum data length 212 indicates the longest receivable data length in the form of bytes for exchanging AKE commands and responses.

Fig. 3C shows the format of the AKE control commands. The algorithm field 201 in the AKE control commands set informs of an executed information exchange procedure when the value of the algorithm ID 200 is 0. Only one bit in the algorithm field 201 of the AKE control command and the response to AKE control commands is set at 1, and the other bits are 0. A bit having the value 1 indicates the information exchange procedure being used. A label 202 is used for identifying correspondence between AKE control commands. For example, let's say a certain information exchange procedure

specifies that the device receiving an AKE control command needs to return a different AKE control command corresponding to the AKE control command received when the AKE control command is sent from one device to another. In this case, the label 202 inserted in the returned AKE control command will have the same value as the label 202

5   inserted in the first AKE control command received, in order to clarify the correlation between both AKE control commands. In step No. 203, a serial number from 1 is given to each AKE control command in the sequence of communication in the information exchange procedure.

A subfunction 299 takes the values shown in Table 1, and the meaning of

10   each AKE command is determined by these values.

Table 1

Subfunction		Value
	Make-response	0016
	Verify-me	0116
	Create-key-	1016
information	Reconstruct-key	1116
	Exchange	2016

If the subfunction 299 is the make-response, this AKE control command

15   challenges the authentication of the device receiving this command. Here, the data 207 contains authentication challenge data expressed as random numbers to authenticate the receiving device. The device receiving this command returns an AKE control command whose subfunction 299 is set to verify-me.

When returning the AKE control command, the data stored in the data 207 is

20   the authentication response data which is a result of a predetermined operation with respect to the authentication challenge data in the received data 207. The key information used for this operation is a key given only to an authorized device in advance. Whether

the device executing the operation is an authorized device or not can be determined by checking the returned authentication response data.

If the subfunction 299 is the create-key-information, this AKE control command requests the encryption key 109 to the device receiving this command. The device receiving this AKE control command returns the AKE control command whose subfunction 299 is set to reconstruct-key. At this point, the encrypted encryption key 109 is stored in the data 207 and returned.

If the subfunction 299 is the exchange, this AKE control command requests the exchange of key information between devices sending and receiving the command. This key information is stored in the data 207 and transferred for indirect authentication between devices or the creation of a common key.

Values other than those specified in Table 1 are reserved for future extension. The channel No. 204 indicates the channel number for isochronous communication between the sending device 110 and receiving device 128. This channel No. 204 is valid only when the subfunction 299 is set to the create-key-information or reconstruct-key. In other cases, this value will be set to FF in hexadecimal format. Block No. 205 and total block No. 206 are used when data which should be handled by the AKE control command cannot be sent by one AKE command. In this case, applicable data is divided into blocks, and transferred in several transmissions. The total block No. 206 indicates the number of divided blocks in applicable data. The block No. 205 indicates the number of each block in the data 207. The data length 209 indicates the valid data length, as bytes, in the data 207. The data 207 is data exchanged by the AKE control command. The device receiving the AKE control command returns a response to that specific AKE control command. The format and value of the response are the same as those of the received AKE control command. The only detail which differs is that the response does not contain the data 207.

Fig. 4 shows a chronological example of AV/C commands which are exchanged between the sending device 110 and receiving device 128 before sending the encryption keys 109 and 126 from the sending device 110 to receiving device 128. First, operations of both devices before exchanging AV/C commands shown in Fig. 4 are

5   briefly described.

An initial condition is that non-encrypted TSP is sent from the sending device 110. The TSP output from the signal source 100 is input to the encrypter 101. Since the output command 105 is set to the normal output, the encrypter outputs TSP as it is without encryption to the source packet generator 102, and registers the value 0 as the

10   encrypting information 910. The source packet generator 102 adds 4-byte source packet header to the TSP received, and outputs it to the CIP block generator 103. The CIP block generator 103 adds 8-byte CIP header 954, and outputs it as isochronous payload 952 to the isochronous packet generator 107. Here, the encrypting information 910 contained in the CIP header 954 is 0 which is input from the encrypter 101. The isochronous packet

15   generator adds the isochronous packet header 900, header CRC 901, and data CRC 903 to the received isochronous payload 952 to create the isochronous packet. This isochronous packet is output to the 1394 bus 111 by the 1394 packet I/O means 108. Since the applicable isochronous packet conforms to the AV protocol, the tag 907 in the isochronous packet header 900 is set to 1.

20   When the TSP output from the signal source 100 is changed, which means that AV information changes from that unprotected AV information to copy-protected AV information, the key generator 106 detects this change, and changes the output command 105 from the normal output to empty output. At the same time, the encryption key 109 for encrypting TSP is given to the encrypter 101.

25   When the output command 105 is for empty output, the encrypter 101 outputs an empty signal to the source packet generator 102 every time it receives a TSP

from the signal source 100, and registers the value 1 as the encrypting information. At receiving the empty signal from the encrypter 101, the source packet generator 102 transmits the received empty signal as it is to the CIP block generator 103 without adding the source packet header. When the CIP block generator 103 receives the empty signal, it

5   outputs only the CIP header 954 to the isochronous packet generator 107. Here, the encrypting information 910 in the CIP header 954 uses the value 1 output from the encrypter 101. The isochronous packet generator 107 creates an isochronous packet as the isochronous payload 952 using the CIP header 954 received from the CIP block generator, and outputs it to the 1394 packet I/O means 108. Since this isochronous packet conforms

10 to the AV protocol, the value of the tag 907 is set to 1. The 1394 packet I/O means 108 outputs received isochronous packet to the 1394 bus 111. This isochronous packet is continuously output, and the isochronous packet only containing the CIP header 954 in this isochronous payload 952 is continuously output to the 1394 bus 111. The receiving device 128 receiving this isochronous packet checks its tag 907 by the 1394 packet I/O

15 means 127, detects that the isochronous packet conforms to the AV protocol, and then outputs this isochronous packet to the payload extractor 123. The payload extractor 123 extracts the isochronous payload 952 from received isochronous packet, and outputs it to the actual data extractor 122. The actual data extractor 122 outputs the encrypting information 910 and source ID 906 in the CIP header 954 to the key generator 125. After

20 the key generator 125 detects that the value of the encrypting information 910 is 1, the receiving device 128 learns that device outputting the isochronous packet from the source ID 906 is the sending device 110. Then, the key generator 125 finally goes onto a process for requesting the encryption keys 109 and 126 using the A/C commands, as shown in Fig. 4.

25   In Fig. 4, the AKE status command 300 is first sent from the receiving device 128 to sending device 110. This enables the receiving device 128 to query

information exchange procedure that can be used by the sending device 110. Replying to this query, the sending device 110 returns the AKE response 301 to the receiving device 128. Information exchange procedure which the sending device 110 can execute is assigned in bits in the algorithm field 201 of the AKE response 301. This allows the

5 receiving device 128 to learn which information exchange procedures can be executed by the sending device. For example, if the sending device 110 can execute the second and sixth information exchange procedures, binary indication in the algorithm field 201 of the AKE response 301 will be 0000000000100010.

The receiving device 128 receiving the AKE response 301 selects one

10 optimal procedure from information exchange procedures that both sending device 110 and receiving device can execute. Then, AV/C commands are exchanged according to the selected exchange procedure. Let's say the receiving device 128 can execute the second and eighth information exchange procedures. Then, the information exchange procedure which can be executed by both sending device 110 and receiving device 128 is only the

15 second procedure. Accordingly, the rest of authentication and information exchange are executed using the second procedure. In the AKE control command in this procedure, the value of algorithm ID will be 0 and the value of the algorithm field 201 will be 0000000000000010 in hexadecimal indication. The information exchange procedure specifies not only the sequence of exchanging a range of AKE control commands but also

20 a format and processing method of the data 207 sent by each AKE control command.

In accordance with the second information exchange procedure, the key generator 125 sends the make response command 302 to the sending device 110. In the data 207 of this make response command 302, two random numbers RRa and RRb generated by the key generator 125 are encrypted, and the algorithm field 201 contains

25 identification information indicating the use of the second procedure. The key used for encryption is a common secret key given to both authorized sending device and receiving

device in advance. The key generator 106 receiving the make-response command 302 checks the algorithm field 201 of the received make-response command 302, and learns to use the second procedure for the rest of authentication and information exchange. Since the key generator 106 can execute the second procedure, the key generator 106 knows that

5   the data 207 of the make response command 302 sent in accordance with this second procedure contains two random numbers encrypted by this secret key. After taking out two random numbers RRa and RRb from the data 207 using this secret key, the key generator 106 returns a response 303 to inform that a response can be generated. Then, the key generator 106 stores one of the random numbers RRa taken out in the data 207, and

10   sends the verify-me command 304 to the receiving device 128. This is the response requested by the previous make-response command 302. Hereafter, the algorithm field 201 of each AKE command exchanged between the sending device 110 and receiving device 128 always contain the identification information indicating the second procedure.

The key generator 125 receiving the verify-me command 304 confirms that

15   RRa in the data 207 conforms to the random number RRa generated by itself, and then returns a response 305 to the verify-me command 304 to inform that verification has completed successfully. The key generator 125 then finally authenticates that the sending device 110 is an authorized sending device.

The sending device 110 then use the make-response command 306 and

20   verify-me command 308 in accordance with the procedures after the make-response command 302 described above to confirm that the receiving device 128 is an authorized receiving device. However, the random number used here is RTa and RTb, and the random number sent back by the verify-me command 308 is RTb.

Now that both sending device 110 and receiving device 128 know the

25   random numbers RRb and RTb, and have confirmed that both are authorized devices, the key generator 106 and key generator 125 separately generates a temporary key (not

illustrated) from RRb and RTb using a common operation method specified by the second procedure. These temporary keys are a common key only between the sending device 110 and receiving device 128.

Next, the key generator 125 sends the create-key-information command 310 to the sending device 110. A channel number of the isochronous packet that the receiving device 128 is currently receiving is stored in the channel No. 204 of the create-key-information command 310. The key generator 106 receiving this create-key-information command 310 encrypts the encryption key 109 to be used for encrypting TSP with the aforementioned temporary key, and then returns a response 311 to inform that the create-key-information command 310 has completed successfully. Then, the key generator 106 sends the reconstruct-key command 312 which stores the encryption key 109 encrypted by the temporary key in its data 207 to the receiving device 128. The key generator 125 uses the temporary key to decrypt the data 207 of the reconstruct-key command 312 received, and obtains the encryption key 126. Then, the key generator 125 returns a response 313 to inform that the reconstruct-key command 312 has completed successfully. Since the encryption keys 109 and 126 are encrypted and decrypted using the same temporary key, they are the same keys. The encryption key 126 is output from the key generator 125 to the decrypter 121. This completes the procedure for granting decrypting information.

The key generator 106 which has sent the reconstruct key command 312 outputs the output command 105 for encrypted output to the encrypter 101. The encrypter 101 receiving this command encrypts TSP received from the signal source 100 by the encryption key 109, and starts to output it to the source packet generator 102. This enables the sending device 110 to send the isochronous packet containing TSP encrypted by the encryption key 109 in its isochronous payload 952 on the 1349 bus 111. This isochronous packet received by the receiving device 128 is decrypted by the decrypter using the

encryption key 126 as described above, and outputs the decrypted packet to the AV generator 120.

In the above series of AKE control commands, each set of the make response command 302 and verify-me command 304; make-response command 306 and verify-me command 308; and create-key-information command 310 and reconstruct-key command 312, respectively has the same label 202. The make-response command 302, verify me command 308, create-key-information command 310, and reconstruct key command 312 also have values 1, 2, 3, 4, 5, and 6 in the step No. 203 respectively.

If the actual data 105 in the isochronous packet output from the sending device 110 changes from encrypted actual data 105 to non-encrypted actual data 105, the decrypter 121 detects the change in the encrypting information 910, stops decryption, and outputs the data received from the actual data extractor 122 as it is to the AV generator 120.

If a bus reset occurs in the 1394 bus 111 after the aforementioned processes shown in Fig. 4 starts, the procedures after and make-response command 302 need to be repeated.

As described above, in the preferred embodiment of the present invention, the sending device sends encrypting information which indicates the encryption status of the actual data in the isochronous packet together with the actual data. This enables the receiving device receiving the isochronous packet to make a request to the sending device for an encryption key for decrypting the actual data if the receiving device detects, by checking the encrypting information in the isochronous packet, that the data is encrypted. The sending device receiving the request then gives the encryption key to the receiving device. Accordingly, the data transfer method of the present invention offers extremely simple procedures for giving the encryption key from the sending device to receiving device for decryption.

Moreover, in the preferred embodiment of the present invention, the isochronous packet transferred via isochronous communication contains i) encrypting information indicating the encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer. This makes possible a data transfer method

5   which has no risk of erroneous operation when a conventional receiving device receives encrypted actual data while maintaining compatibility with the conventional data transfer method.

Furthermore, in the preferred embodiment of the present invention, the CIP header remains non-encrypted for transfer even if encryption of synchronous data starts

10   while the receiving device is continuously receiving synchronous data sent by the sending device. This enables a data transfer method which eliminates the possibility of erroneous operation of the receiving device receiving the data.

In the preferred embodiment, once encryption by the encryption key starts, actual data in all transfer units is encrypted and sent. However, it is not necessary to

15   encrypt all units of data to be transferred. For example, even if both encrypted transfer units and non-encrypted transfer units are sent alternately, the receiving device can correctly decrypt the data because encrypting information is included in the CIP header, thus achieving the same effect. In addition, it is apparent that the same effect is also achievable even if the receiving device specifies a percentage of encrypted transfer units

20   to the sending device. The size of the MPEG source packet is 192 bytes, with more than one source packet stored in one isochronous payload in the case of the high data rate transfer of MPEG (12 Mbps minimum). Naturally, however, it is not possible to have both encrypted source packet and non-encrypted source packet in the same isochronous payload.

25   In the preferred embodiment, all actual data is encrypted using the encryption key. However, it is not necessary to encrypt all pieces of data. For example,

the same effect is achievable by encrypting the first half of the actual data, or encrypting the first and third quarters of the actual data. In this case, the receiving device can decrypt appropriately, if, when sending the data, information is inserted to indicate encrypted portions and their percentage in the CIP header. The same effect is also achievable by

5   inserting in the CIP header encrypting information announcing whether the actual data is encrypted or not. The receiving device queries the sending device via asynchronous communication about which part of the actual data is encrypted and to what level, when the receiving device detects encryption by checking the CIP header. The same effect is also achievable in this case even when the receiving device specifies the encryption area

10   and percentage to the sending device via asynchronous communication. If only the confidential portion in the actual data is encrypted, the burden for encryption and decryption is reduced, and at the same time, a sufficient effect of encryption may be achieved.

In the preferred embodiment, the isochronous packet containing only the

15   CIP header without actual data is transferred until the completion of mutual authentication between the sending and receiving devices. However, the same effect is achievable even when an isochronous packet containing encrypted actual data is output from the start, and not the isochronous packet containing only the CIP header.

In the preferred embodiment, procedures for transferring the AKE control

20   commands between sending and receiving devices are determined by mutual negotiation. However, if the receiving device features only one executable procedure, the same effect is achievable by starting to transfer commands immediately, without executing this negotiation procedure, using only the executable procedure. In this case, it may be preferable to specify in advance a basic minimum of executable procedures for all

25   authorized devices.

In this preferred embodiment, direct authentication is implemented between the sending and receiving devices, following which decrypting information is transferred using a secret key. However, the means for transferring authentication and decrypting information is not limited to this procedure. For example, a public key may be used for

5 mutual indirect authentication and the creation of a temporary key. Decrypting information may then be transmitted using this temporary key. Such procedures are briefly described below.

The sending and receiving device stores the key information necessary for mutual indirect authentication in the data 207 of the AKE control commands, and send

10 this information to each other in line with a procedure determined by mutual negotiation. Here, the subfunction 299 is set to the exchange. This enables both sending and receiving devices to share the same temporary key if they are both authorized devices. Decrypting information is then transferred using the create-key-information command and reconstruct-key command in accordance with the same procedures as those described in

15 the preferred embodiment.

In the preferred embodiment, the procedure for transferring AKE control commands exchanged between sending and receiving devices is determined by mutual negotiation. If the types of procedures executable by the sending device are known in advance, the same effect is achievable by having the receiving device transfer commands

20 using a procedure executable by the sending device without first executing this negotiation procedure.

In the preferred embodiment, procedures for transferring AKE control commands exchanged between sending and receiving devices are determined by mutual negotiation. However, the method for determining transfer procedures is not limited to

25 this one. More specifically, if priority is given in advance to each of several transfer procedures, the receiving device may start to transfer using the procedure given the

highest priority which is executable by itself. If the sending device cannot execute that procedure, the receiving device tries to transfer data by going down the list of procedures in order of priority until a procedure that is executable by both the sending and receiving device is found. The AKE control commands are then transferred using this procedure to

5   achieve the same effect.

In the preferred embodiment, the sending device encrypts decrypting information which is used for decrypting actual data before transferring it to the receiving device. However, the way the receiving device obtains decrypting information is not limited to this procedure. In other words, the sending device may provide the receiving

10   device sufficient information for obtaining decrypting information, without transferring encrypted decrypting information, and the receiving device may obtain decrypting information indirectly from this information. More specifically, the sending device transfers only the type of hash function to the receiving device, and the receiving device obtains decrypting information using the received type of hash function to achieve the

15   same effect.

The preferred embodiment described above comprises an example of the AKE command format. However, the AKE command format is not limited to this one. In other words, the AKE command format indicated in this embodiment is just one example of how the preferred embodiment may be realized. The same effect is achievable by using

20   commands in a different format.

Industrial applicability

As described above, the present invention has the significant effect of realizing a data transfer method using extremely simple procedures for passing key

25   information from the sending device to the receiving device for decryption. Encryption identification information indicating the encryption status of actual data in synchronous

data is sent together with actual data. The receiving device receiving the synchronous data checks the encryption identification information in the synchronous data, and if it detects that the actual data is encrypted, the receiving device requests the sending device for decrypting information for decrypting the encrypted data. The sending device receiving

5   this request gives the decrypting information to the receiving device.

The present invention has another significant effect of realizing a data transfer method which eliminates the possibility of erroneous operation of the receiving device even if conventional receiving device receives encrypted synchronous data, while maintaining compatibility with a conventional data transfer method. Synchronous data

10   transferred through synchronous communication contains i) encryption identification information indicating encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer.

The present invention has still another significant effect of realizing a data transfer method which eliminates the possibility of erroneous operation of the receiving

15   device even if encryption of synchronous data starts while the receiving device continuously receives synchronous data sent from the sending device. Synchronous data transferred through synchronous communication contains i) encryption identification information indicating encryption status of the actual data and ii) actual data, but only the actual data is encrypted for data transfer. This enables to transfer the CIP header as it is

20   without being encrypted.

The present invention has still another significant effect of realizing a data transfer method which always executes the most suitable procedure even when new and conventional devices share the same network. A procedure for both authentication and decrypting information with good future extendibility are achievable by selecting a

25   procedure for providing authentication information and decrypting information exchanged between the sending and receiving devices by negotiation between the sending and

receiving devices. In other words, even if a new authentication method or decrypting information become available in the future, the most suitable procedure will remain selectable by negotiation between devices even if a device which can use the new procedure and a device which can use only conventional procedures share the same

5   network, as long as the new device is back-compatible with older procedures.

The present invention has still another significant effect of realizing decryption even if software which has a low encryption/decryption processing rate is used. The present invention allows the relative proportion of encrypted actual data and non-encrypted actual data to be varied within a single file. Accordingly, even if the

10   receiving device has no exclusive hardware for high-speed data decryption, software can be used instead. More specifically, even if the receiving device has no hardware for decryption like PC, rapid processing is made possible by reducing the proportion of encrypted data in the file and thus shortening the time required for the decryption process.

The present invention has still another significant effect of realizing a data

15   transfer method which uses the limited bus transfer band efficiently and significantly reduces the risk of unauthorized device receiving readable data. Unless the sending and receiving devices mutually authenticate that both are authorized devices, isochronous packets without actual data are output.